2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

# PLL-based Pitch Detection and Tracking for Audio Signals

Udo Zölzer, Sujaritha Vettukadu Sankarababu, and Stephan Möller
*Helmut Schmidt University - University of the Federal Armed Forces*
*Hamburg, Germany*
*Email: udo.zoelzer@hsu-hamburg.de*

***Abstract*—Pitch detection and tracking can be used for several audio effects, such as time and pitch scaling and the generation of harmonic signals related to the detected fundamental frequency. A variety of audio effects has efficiently been implemented by time-domain processing. Pitch tracking of monophonic audio signals can also be performed by time-domain techniques. This paper introduces a novel phaselock loop based frequency demodulator for pitch detection and tracking. We will derive the guidelines for parameters of such a kind of pitch tracker and show performance evaluations for a certain type of input signals. The resulting implementation is suited for producing real-time pitch tracking on simple hardware systems with low computational complexity.**

***Keywords*-PLL, phaselock loop, pitch, detection, tracking**

## I. Introduction

Several audio applications need the pitch of the audio signal for further processing steps. The extraction of the pitch on a sample by sample basis is the main motivation for exploring efficient and robust techniques with low computational complexity. Simple time-domain approaches are based on oscillators and phaselock loops (PLLs) control oscillators in an adaptive manner [1]. These PLLs have proven to be reliable frequency trackers and synchronizers for a variety of applications and mainly in the field of communications engineering. Especially for audio signals, where time-varying pitches occur such as linear and quadratic increasing/decreasing pitches, higher order phaselock loops are required. For pitch tracking of speech a bank of PLLs is described in [2]. Pitch detection and tracking of audio signals based on a single feedback frequency demodulator is introduced in [3]. Here, we will extend this kind of pitch tracking and derive a novel third-order PLL which operates in a nonlinear mode with a special loop filter for the fundamental frequency (pitch) computation. We give straight forward design rules for all relevant parameters. The resulting implementation has proven to be well suited for real-time pitch tracking on simple hardware systems with low computational complexity. In section II, we give an introduction into the building blocks of a digital PLL for pitch detection and tracking of audio signals. In section III, we discuss the design of a special new loop filter for the PLL. Finally, we evaluate the performance of the pitch tracker for artificial and real-world audio signals in section IV and summarize the results in conclusion.

## II. PLL Design

The design of a PLL for audio applications is based on the frequency range of operation and the tracking ability for time-varying input frequencies. An upper frequency limit of 1.5 kHz is an obvious choice for audio signals. Furthermore, the amplitude variation of the input audio signal should be removed by an automatic gain control. This type of preprocessed signal with a nearly constant time-domain envelope is then fed to the PLL as input. The fundamental frequency $F0(n)$ detection and tracking is performed by the PLL system shown in Fig. 1. First, the input signal is band-limited to a frequency band of 20 Hz up to 1.5 kHz by a sixth-order low-pass and a second-order high-pass filter. Next, a constant envelope processing according to

$$x_{in}(n) = [x(n) * h_{LP6th}(n) * h_{HP2nd}(n)] \cdot g_{env}(n) \quad (1)$$

is performed by multiplying the band-limited input by an envelope factor $g_{env}(n) = 1/x_{env}(n)$. This envelope factor is the inverse of the time-domain envelope $x_{env}(n)$ of the filtered input signal. The novel third-order PLL is then used for F0 tracking of the audio signal. The constant envelope signal $x_{in}(n)$ multiplied by a feedback oscillator output $y_{osc}(n)$ gives

$$x_d(n) = [x_{in}(n) \cdot y_{osc}(n)] \cdot K_d, \quad (2)$$

where $K_d$ is the loop gain of this feedback system. This loop gain coefficient controls the frequency range of detection and tracking. Inside the loop a special second-order loop filter defined by

$$\begin{aligned} F_{LF}(z) &= 1 + H_{LP2nd}(z) & (3)\\ F_{LF}(z=1) &= 1 + 1 = 2 & (4)\end{aligned}$$

delivers a signal $f_{osc}(n)$ and the fundamental frequency $F0(n)$ given by

$$\begin{aligned} f_{osc}(n) &= x_d(n) * f_{LF}(n) & (5)\\ F0(n) &= [x_d(n) * h_{LP2nd}(n)] \cdot 2. & (6)\end{aligned}$$

The factor 2 in Eq. (6) compensates for the DC gain of 2 in Eq. (4). The feedback oscillator defined by

$$y_{osc}(n) = \cos\left[2\pi \frac{f_{osc}(n)}{f_S} n\right] \quad (7)$$

978-0-7695-4712-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IIH-MSP.2012.110

IEEE computer society

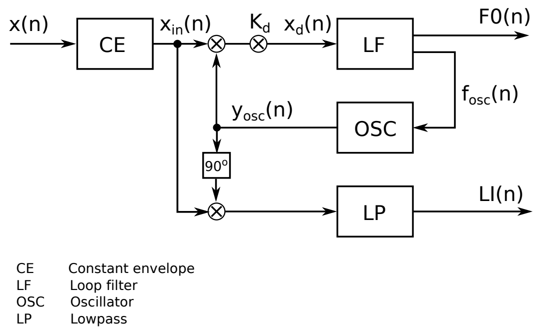


Figure 1. PLL-based pitch tracker with special loop filter and lock indicator.

is controlled by the output of the loop filter denoted by $f_{osc}(n)$ and given by Eq. (5) (plus quiescent frequency $f_q$ for starting the PLL operation, if necessary). This feedback oscillator can be implemented as a recursive complex-valued filter with quadrature output signals. The cosine part given by Eq. (7) is used for delivering $F0(n)$ and $f_{osc}(n)$ in the upper branch of Fig. 1. The sine part of the oscillator can be used for multiplying with $x_{in}(n)$ and subsequent filtering to derive a lock indicator signal $LI(n)$ [1].

## III. Loop filter

The new loop filter is a combination of a direct path and a second-order low-pass filter, which together form a low-frequency shelving filter

$$F_{LF}(z) = 1 + H_{LP2nd}(z) \quad (8)$$

$$= \frac{1 + b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}. \quad (9)$$

This parallel second-order low-pass filter (in comparison to the integrator like parallel branches) is different from published PLL loop filters, and offers parameterized and robust tracking for audio signals. In the PLL literature [1], the open-loop transfer function is denoted by $G(z)$, the closed-loop transfer function by $H(z)$ and the error transfer function by $E(z)$ and can be written as

$$G(z) = K_d F_{LF}(z) \frac{z^{-1}}{1 - z^{-1}} = K_d F_{LF}(z) \frac{1}{z-1}, \quad (10)$$

$$H(z) = \frac{G(z)}{1 + G(z)} = \frac{K_d F_{LF}(z)}{(z-1) + K_d F_{LF}(z)}, \quad (11)$$

$$E(z) = \frac{1}{1 + G(z)} = \frac{z-1}{(z-1) + K_d F_{LF}(z)}. \quad (12)$$

The transfer function $E(z)$ is the complementary transfer function to $H(z)$, both sum up to $H(z) + E(z) = 1$. $H(z)$ is a low-pass filter, and therefore $E(z)$ will be a complementary high-pass filter. The closed-loop transfer function with the special loop filter $F_{LF}(z)$ given by Eq. (8)-(9) is finally evaluated as

$$H(z) = \frac{K_d(1+b_0+b_1 z^{-1}+b_2 z^{-2})}{(z-1)(1+a_1 z^{-1}+a_2 z^{-2})+K_d(1+b_0+b_1 z^{-1}+b_2 z^{-2})} \quad (13)$$

which leads to a third-order PLL because of the third-order transfer function for $H(z)$. The error transfer function is the complementary transfer function given by

$$E(z) = \frac{(z-1)(1+a_1 z^{-1}+a_2 z^{-2})}{(z-1)(1+a_1 z^{-1}+a_2 z^{-2})+K_d(1+b_0+b_1 z^{-1}+b_2 z^{-2})}. \quad (14)$$

This error transfer function has one zero at $z_{0,1} = 1$ and two complex-conjugate zeros $z_{0,2/3} = z_{\infty,1/2}$ at the original pole locations of the low-pass filter $H_{LP2nd}(z)$. This filter acts as a high-pass filter for the phase error after the phase detector, which is implemented as a multiplier. The design of this novel third-order PLL is completely determined by the three parameters:

1) loop gain $K_d$ which controls the frequency range of the PLL by adjusting the cut-off frequency of the filter $H(z)$ (given by Eq. (13)). The frequency responses of $H(z)$ and $E(z)$ for varying $K_d$ are shown in Fig. 2.
2) cut-off frequency $f_c$ of the second-order low-frequency shelving filter $F_{LF}(z)$ (controls rise time of $F0$ step, see Fig. 3),
3) and $Q$ factor of the second-order low-frequency shelving filter $F_{LF}(z)$ (controls the tracking curve, as shown in Fig. 3).

For completeness, the filter coefficients of the second-order low-pass filter inside the loop filter are given by

$$a_1 = 2 \cdot (k^2 - 1) \cdot k_1 \quad (15)$$

$$a_2 = (1 - k/Q + k^2) \cdot k_1, \quad (16)$$

$$b_0 = k^2 \cdot k_1 \quad (17)$$

$$b_1 = 2 \cdot k \cdot k_1 \quad (18)$$

$$b_2 = k^2 \cdot k_1 \quad (19)$$

with $k = \tan(\pi \frac{f_c}{f_S})$ and $k_1 = \frac{1}{1+k/Q+k^2}$. Figure 3 shows an input signal $x_{in}(n)$ and the extracted pitch $F0(n)$ for three different low-pass filters with varying $Q$. For $Q = 1/\sqrt{2}$ and $Q = 1/2$, an overshoot of the pitch contour can be observed in comparison to $Q = 1/3$ without any overshoot. This is demonstrated by the third plot showing the beginning of the signal and the fourth plot showing the frequency jump from 800 Hz down to 80 Hz. A frequency ripple is occuring around the fundamental frequency which is due to the second harmonic of the input frequency. This ripple can be further reduced by appropriate filtering of the $F0$ signal, or by choosing a lower cut-off frequency of the low-pass filter, which consequently leads to a longer attack time. The tracking ability of PLLs can be evaluated analytically by considering the lock state of the PLL and then using final value theorems of the z-transform [1] to monitor the phase

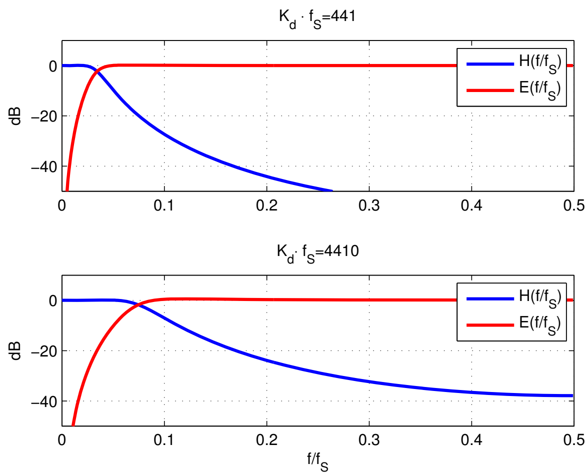


Figure 2. Frequency responses for $H(z)$ and $E(z)$ for varying $K_d$.

behavior after the multiplier (the so-called phase detector). The derivative of the phase gives the instantaneous frequency which we detect with the PLL. We can write the following z-transforms for a phase step $s(n)$, a phase ramp $r(n)$, and a phase acceleration $a(n)$:

$$s(n) = \epsilon(n) \rightarrow \frac{z}{z-1}, \tag{20}$$

$$r(n) = n\epsilon(n) \rightarrow \frac{z}{(z-1)^2}, \tag{21}$$

$$a(n) = n^2\epsilon(n) \rightarrow \frac{z(z+1)}{(z-1)^3}. \tag{22}$$

The final values for the phase errors for a phase step and a phase ramp can be estimated by

$$\begin{aligned} p_s(n)|_{n\rightarrow\infty} &= \lim_{z\rightarrow 1}(z-1)E(z)\frac{z}{z-1} \\ &= 0 \end{aligned} \tag{23}$$

$$\begin{aligned} p_r(n)|_{n\rightarrow\infty} &= \lim_{z\rightarrow 1}(z-1)E(z)\frac{z}{(z-1)^2} \\ &= \frac{1+a_1+a_2}{K_d(1+b_0+b_1+b_2)} \\ &= \frac{1}{2K_d} \end{aligned} \tag{24}$$

where $E(z)$ is given by Eq. (14). For the phase acceleration we get

$$\begin{aligned} p_a(n)|_{n\rightarrow\infty} &= \lim_{z\rightarrow 1}(z-1)E(z)\frac{z(z+1)}{(z-1)^3} \\ &= \lim_{z\rightarrow 1}\frac{(1+a_1z^{-1}+a_2z^{-2})}{D(z)}\frac{z(z+1)}{(z-1)} \end{aligned} \tag{25}$$

where $D(z)$ is the denominator of Eq. (14). Considering the resulting term in Eq. (25), one can notice a singularity. But, the numerator has two complex-conjugate zeros close to $z = 1$, which counteract the remaining pole in the denominator of Eq. (25). Taking into account that frequency ramps and even quadratic increasing frequencies are limited by an upper frequency bound, in practice it is sufficient that the PLL tracker should follow such increasing frequency contours, but they will never go to infinity. These simplified inputs offer just a first insight on the behavior of the used PLL. In the course of the development of the third-order PLL a real guitar input signal has been used to explore the behavior of the $F0$ responses with parallel real-time measurements [3].

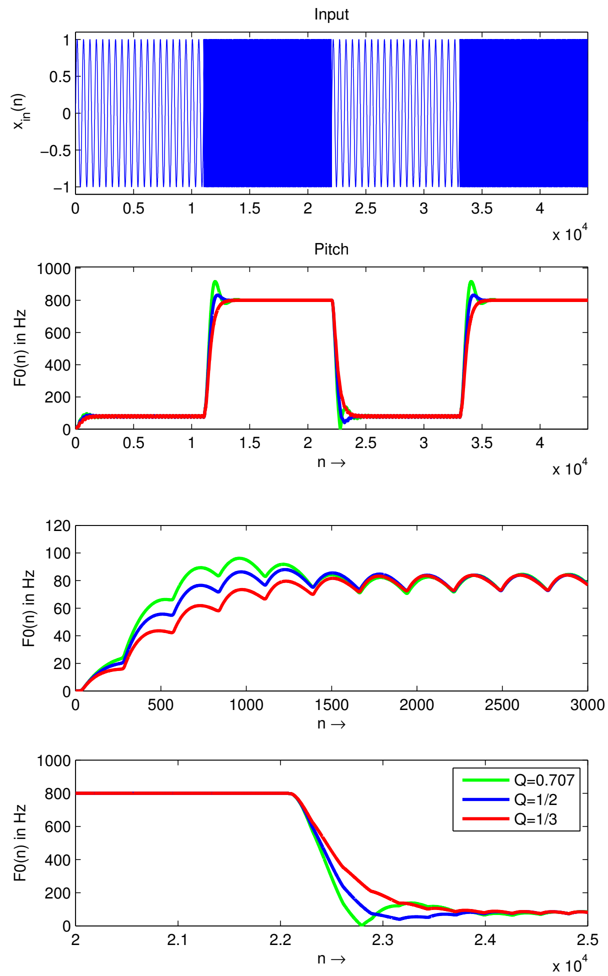


Figure 3. Input signal $x(n)$ and detected $F0(n)$ with different low-pass filters ($Q = 1/\sqrt{2}, 1/2, 1/3$ and $f_c = 20$ Hz).

## IV. Results

In this section, selected test signals are used to demonstrate the detection and tracking behavior of the PLL. Figure 4 shows a real-world guitar signal, its corresponding spectrogram, and the extracted $F0$. The extracted pitch contour follows the played tune quite well and falls down,

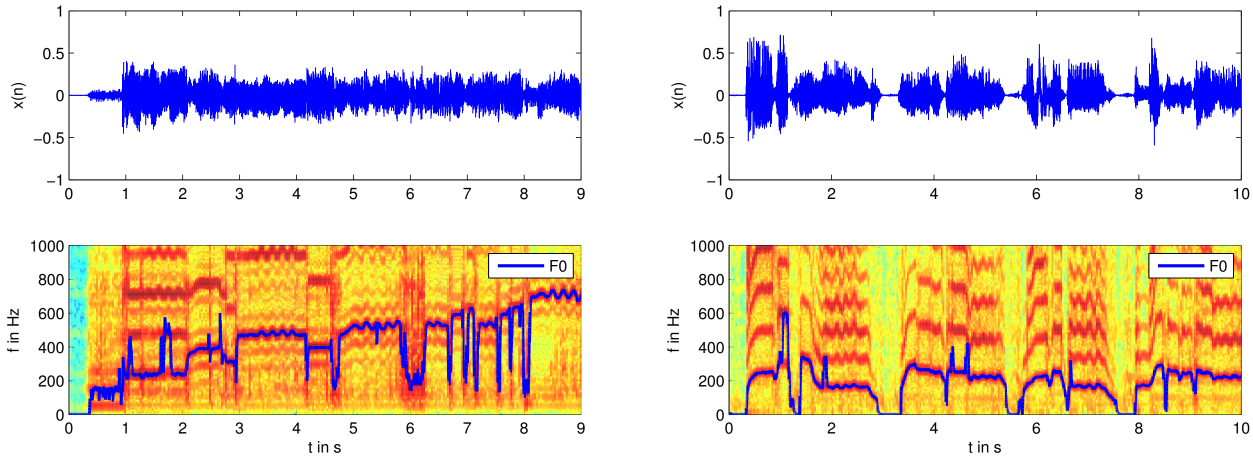


Figure 4. Guitar (left) and vocal (right) signals, spectrograms, and detected $F0$ contours.

if pauses occur during playing. One can notice that a frequency jump between $t = 1$s to $t = 2$s happens, when the fundamental frequency is weaker than the harmonic frequencies. Moreover, during the attack durations of played notes, the tracker shows a transient behavior, which remains unsuppressed as shown in the resulting pitch curve. String bendings and vibratos at the end of longer notes are tracked properly. As a further example, a vocal signal from a male singer is shown in the right part of Fig. 4 with its corresponding spectrogram and the extracted pitch contour. Essential for a robust operation with real-world signals is the constant envelope processing in front of the PLL. Here a proper envelope processing plus adaptive filtering is important. Several periodic chirp signals are shown in Fig. 5. The $F0(n)$ contours show a proper detection and tracking ability of the PLL. For fast linear and quadratic chirp signals, the tracking is sufficiently accurate. In a real-time implementation on a fixed-point DSP processor, the PLL-based pitch tracker shows excellent performance for guitar inputs. The adaption of parameters for vocals or other instruments is currently under investigation.

## V. Conclusions

The PLL-based pitch tracker offers a sample by sample instantaneous frequency estimation (pitch or fundamental frequency) with low computational complexity. The pitch signal can be used for controlling parameters for further signal processing steps. Even in a block-based signal processing approach this kind of pitch tracker allows instantaneous frequency estimation inside a block. The parameters of the third-order PLL are easily adjustable to the special audio application.

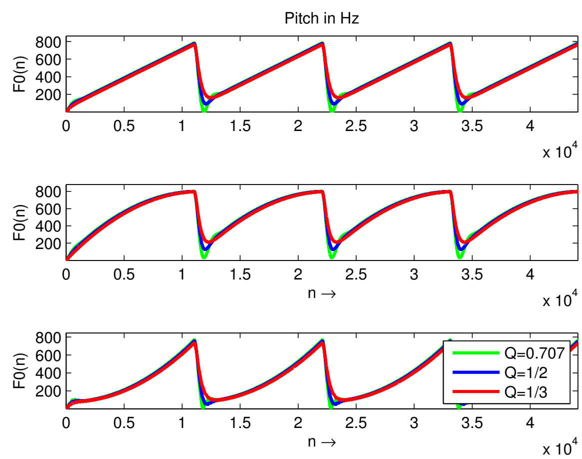


Figure 5. Periodic ($T = 0.25$s) linear, quadratic concave and quadratic chirps from 80 to 800 Hz.

## References


[1] Floyd M. Gardner, *Phaselock Techniques*, J. Wiley & Sons, Third edition, 2005.

[2] P. Pelle, C. Estienne, and H. Franco, "Robust Speech Representation of Voiced Sounds based on Synchrony Determination with PLLs," in *Proc. IEEE ICASSP-2011, Prague*, May 2011, pp. 5424–5427.

[3] U. Zölzer, "Pitch-based Digital Audio Effects," in *Proc. 5th Int. Symposium Communications, Control and Signal Processing (ISCCSP), Rome*, May 2012.